

GitHub Campus Advisors

Teacher training to master Git and GitHub



GitHub Education

Module 2

Individual work

Network activity

Remotes

Fetch/push

Branches

Managing student repositories

GitHub Classroom

Permissions

Exercises

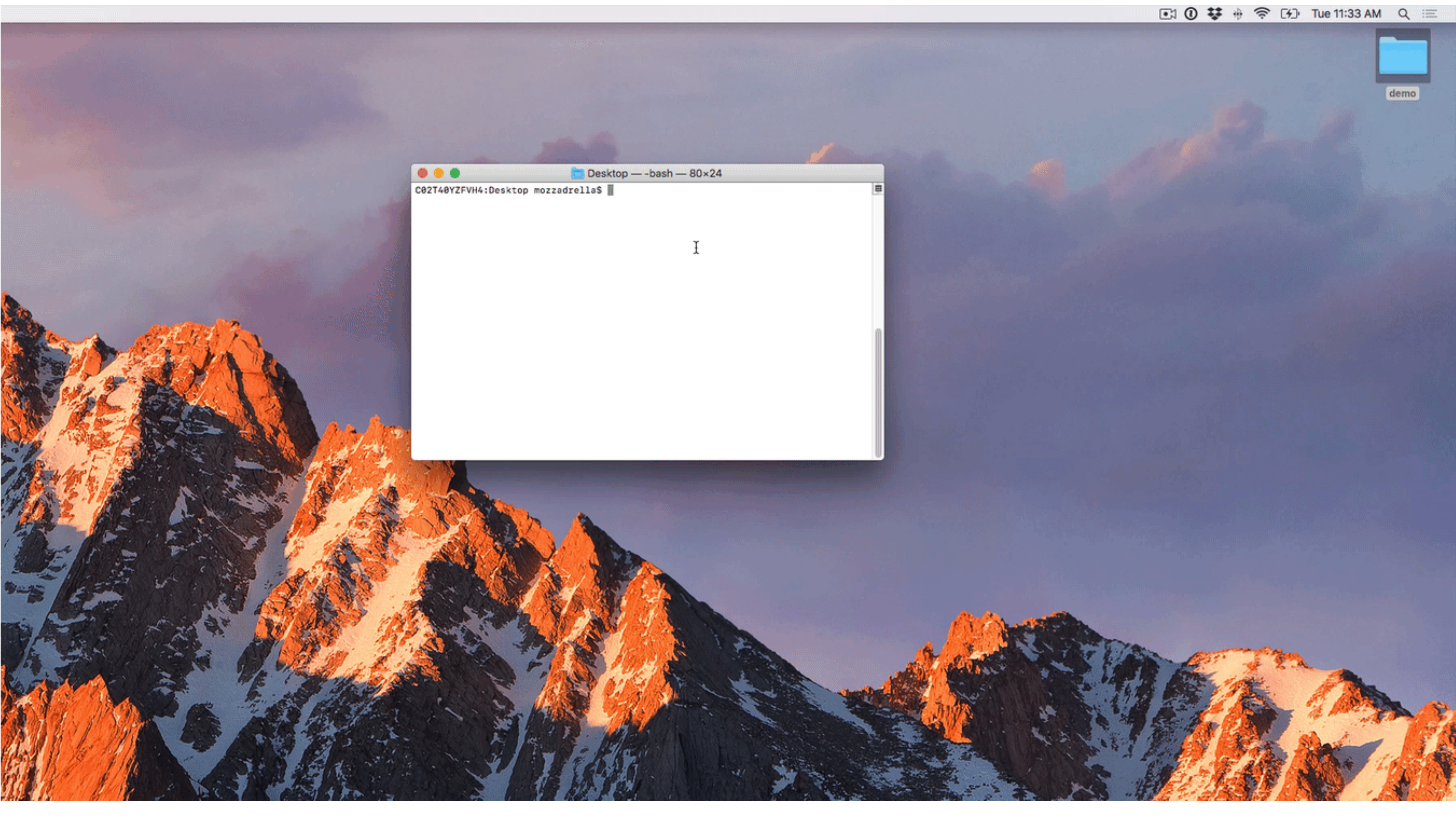


Individual work

Network activity



```
Desktop — -bash — 80x24
C02T40YZFVH4:Desktop mozzadrella$
```



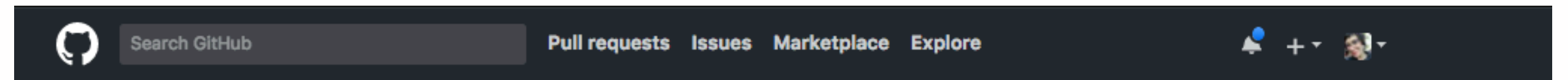
GitHub

- Hosts your repositories
- Track student progress
- Social features to enable collaboration



**Let's create a repository in
GitHub!**

Let's set up a place to host your code




- A repository on GitHub!
- <https://github.com/new>
- As a teacher, you have access to free private repositories, but let's choose public for now


Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 mozzadrella

Repository name

individualassignment 

Great repository names are short and memorable. Need inspiration? How about [didactic-invention](#).

Description (optional)

Individual Assignment for Campus Advisors

Public

Anyone can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None 

Create repository



**Adding a remote
allows the transfer of
your commits to
another machine.**



Hello! I would like to
send you my recent
commits.



The bookmarked location is referred to as a "remote."



Hello! I would like to send you my recent commits.

```
git remote add origin (REPO LOCATION)
```



Add origin

Send my commits to a location.



And origin is at this address.



```
git remote add origin (REPO LOCATION)
```



This statement names the remote "origin."



Back to the terminal!

Pushing to a remote



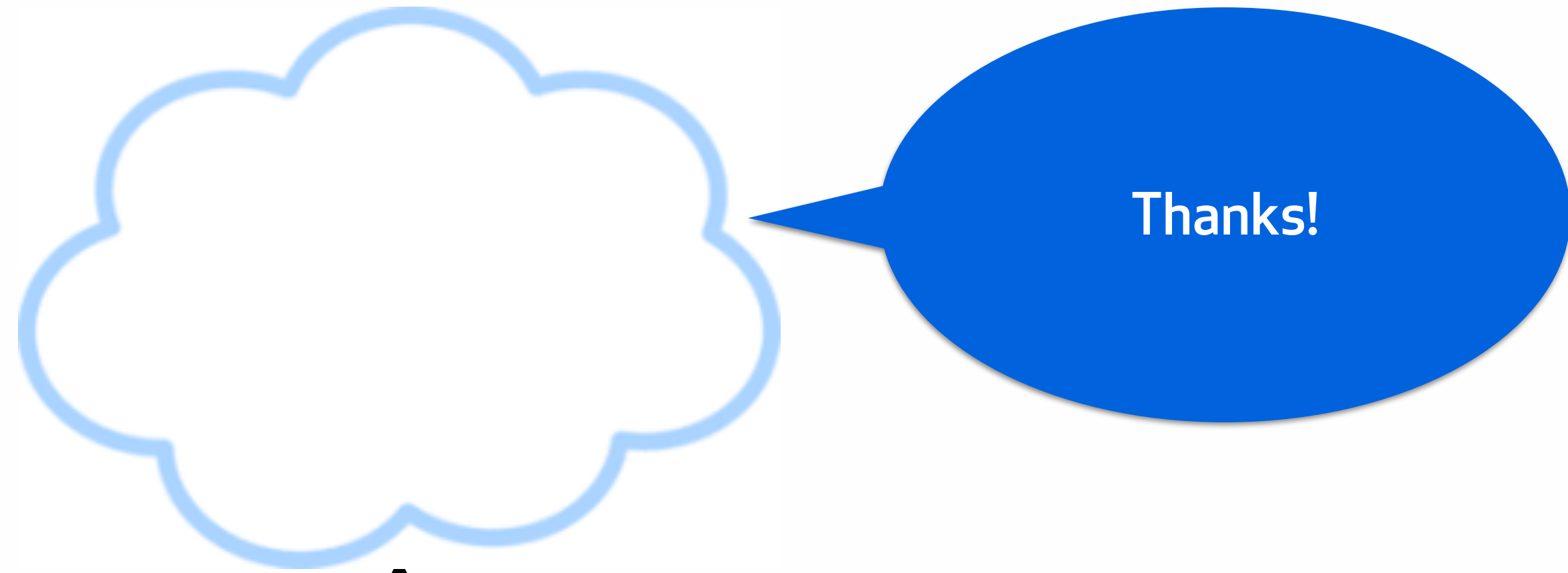
How do you get your commits up to the remote?



```
git push origin master
```



How do you get your commits up to the remote?



```
git push origin master
```



Link remote with local.

-u is short for --set-upstream

```
git push -u origin master
```



Useful because you can just write "git push" when you want to push future commits.



Types of remote addresses

- HTTP/HTTPS urls
- Git protocol over SSH and use the file path
- GitHub Desktop client (clone repository and open in Desktop)



Check remotes with -v

Where am I sending and receiving commits from?



```
C02T40YZFVH4:demo mozzadrella$ git remote -v  
origin https://github.com/mozzadrella/demo (fetch)  
origin https://github.com/mozzadrella/demo (push)  
C02T40YZFVH4:demo mozzadrella$
```



**Now: editing in the GitHub UI
(not recommended)**

Create a New Repository

GitHub, Inc. [US] | https://github.com/new

remove-an-inheritan... Switch to GitHub:lab Rails 3.2.22.5 2.1s

Search GitHub Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mozzadrella / Repository name:

Great repository names are short and memorable. Need inspiration? How about `didactic-invention`.

Description (optional): Individual Assignment for Campus Advisors

- Public**
Anyone can see this repository. You choose who can commit.
- Private**
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Create repository

Desktop/individual

gobs.md

al Assignments: Learning Objectives

demo

LF UTF-8 GitHub Markdown master 0 files

Fetch



I'd like the commits I don't already have, please.



Fetch



git fetch



```
-bash: /Users/mozzadrella/Desktop/individual: is a directory
[C02T40YZFVH4:individual mozzadrella$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mozzadrella/individual
 22a8808..c0c7653 master    -> origin/master
C02T40YZFVH4:individual mozzadrella$ █
```



Fetch

Hmm, when I run git log I can't see these commits in my local repo. :(



git fetch



```
-bash: /Users/mozzadrella/Desktop/individual: is a directory
[C02T40YZFVH4:individual mozzadrella$ git fetch
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mozzadrella/individual
 22a8808..c0c7653 master    -> origin/master
C02T40YZFVH4:individual mozzadrella$ █
```

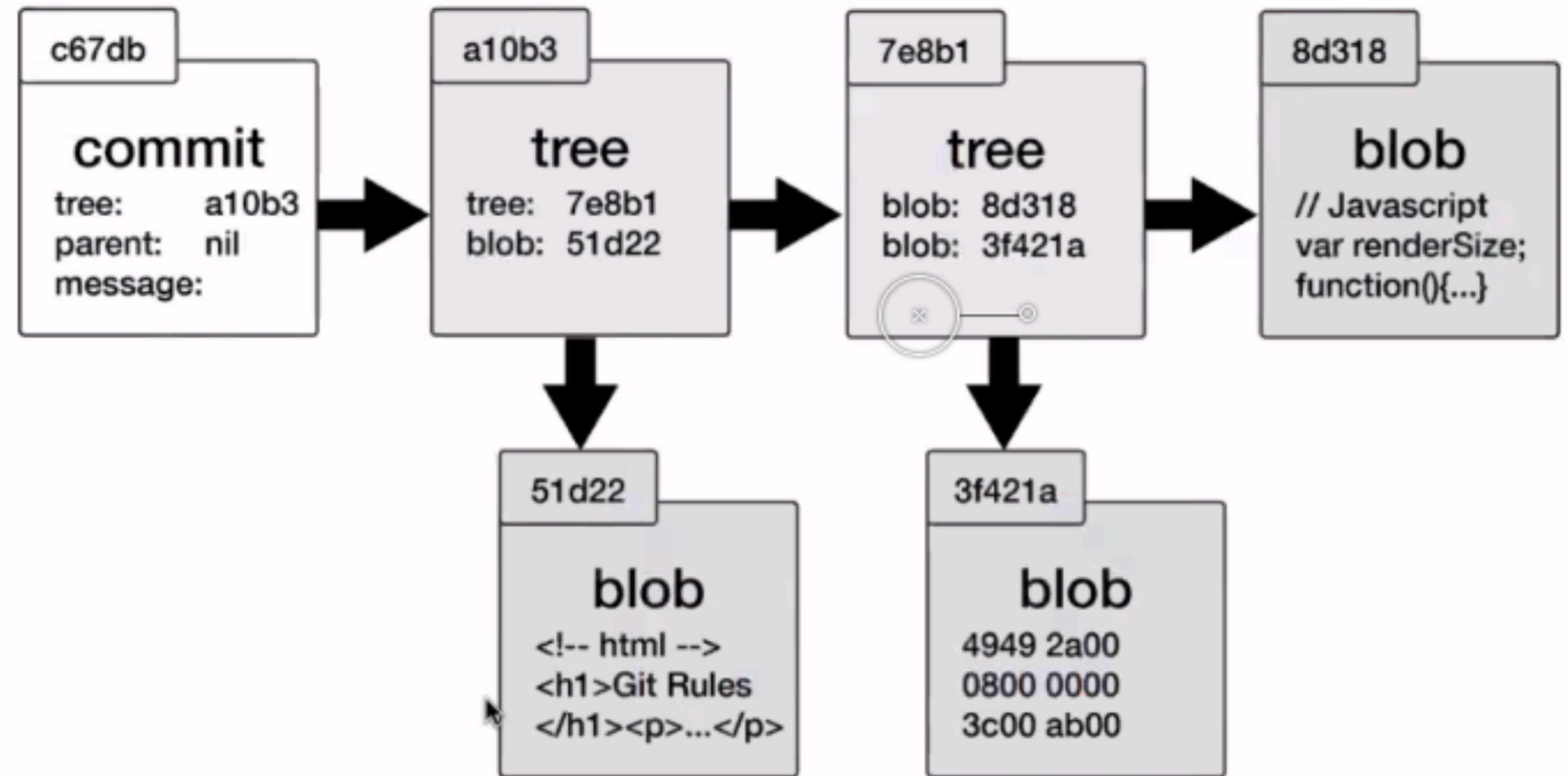


Counting objects

Git **only** transmits the necessary objects.

Push: sends objects the remote doesn't have.

Fetch: receives objects we don't have locally.



Work with remotes



Work with remotes

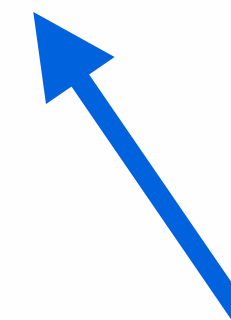
- On the command line: create a repository from the command line called "individual-work."
- On GitHub.com, create a repository.
- On GitHub.com, upload your first week's assignments to the "Individual" repository.
- Use the command line to bring the commits back down to your local repository.



Which branch do you want to push?



```
git push -u origin master
```



You want to push master.
To origin, the remote.



But what is “master”?



```
git push -u origin master
```





**You've been on a branch...
all along.**



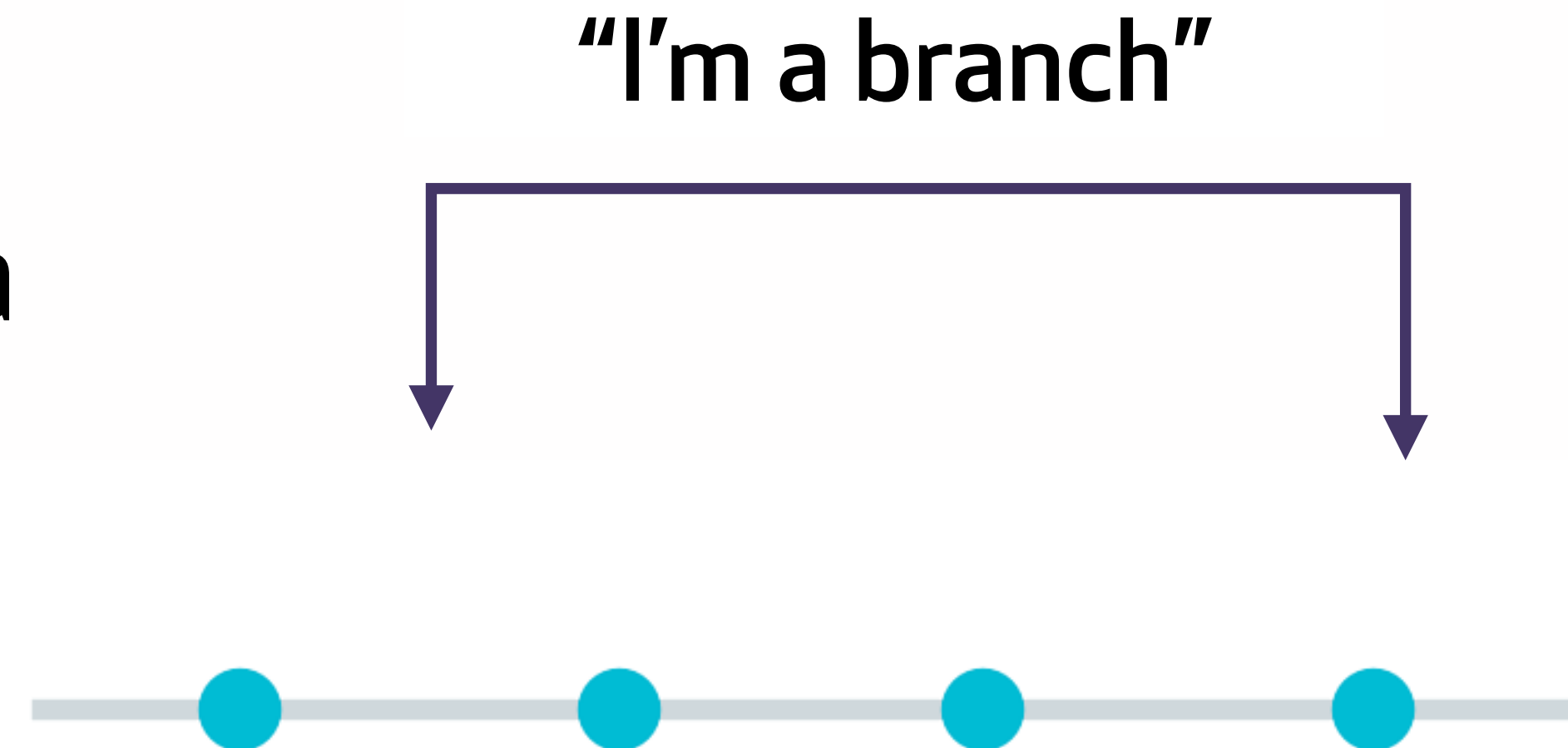
**An aside to discuss
branches.**

Branches are bookmarks to commits

“Master” is the default, it’s a naming convention.

Can think about branches as either a bookmark or a pointer for commits.

As we add commits, the active branch updates to point to the newest commit (HEAD).



demo : master

demo

- Changes
- All Commits
- Branches
 - master 3↑
- Remotes
 - origin
 - master
- Tags
- Stashes
- Submodules


• master add info about Fork, m...

• add info about branches

• add learning objectives to readme.

• origin/master readme.md

Commit Changes File Tree




 Author: Mozzadrella <mozzadrella@github.com>

 Date: December 16, 2017 at 1:28:05 PM EST

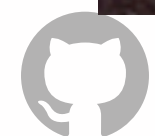
 Commit hash: cc30f26f9bcb27fc45338961a3f09b269ecd0931

 Parents: [6c8bfc8](#)

Subject: add info about Fork, my handy git client

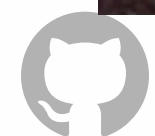
Changes:  readme.md

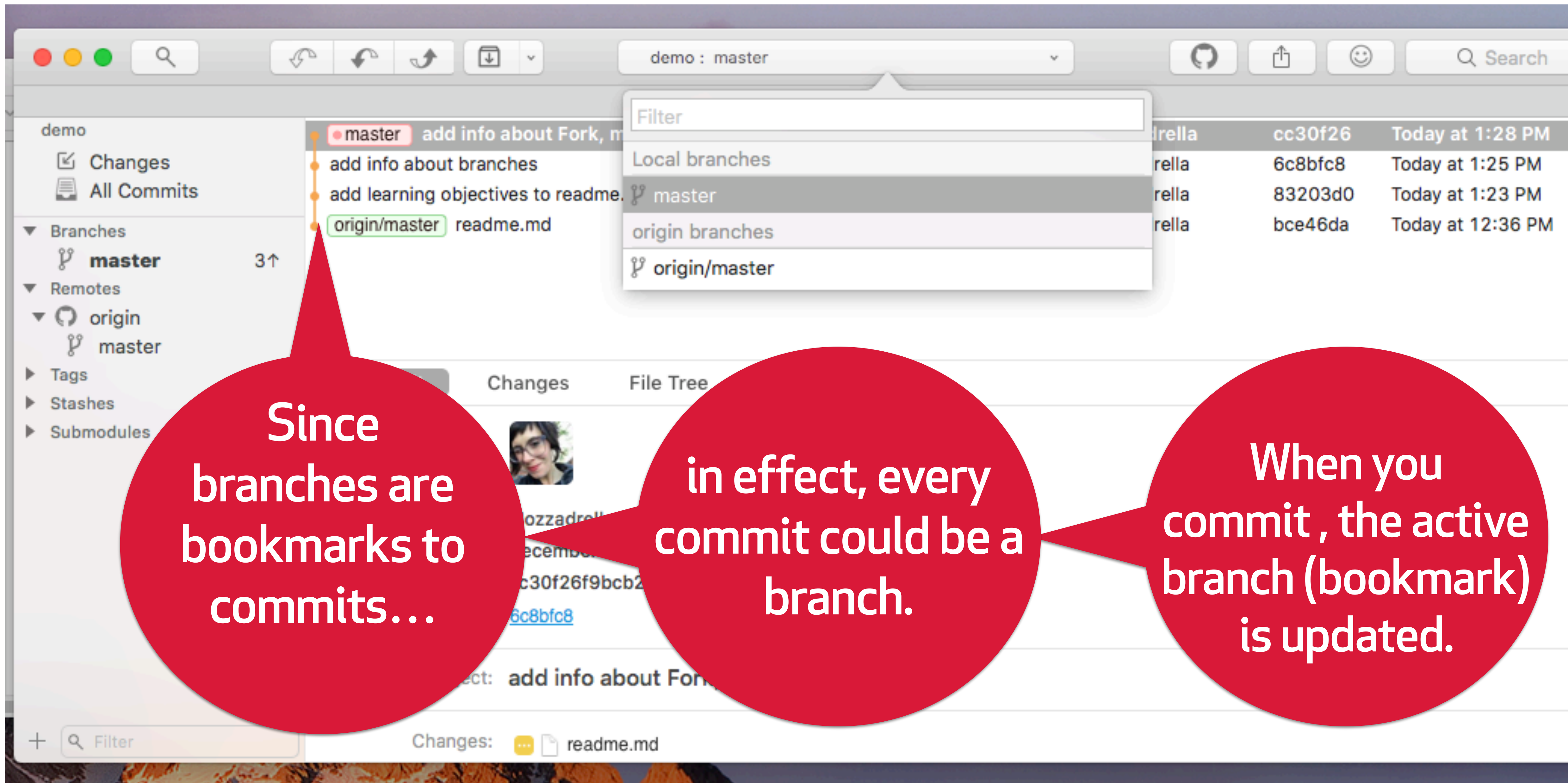
irella	cc30f26	Today at 1:28 PM
irella	6c8bfc8	Today at 1:25 PM
irella	83203d0	Today at 1:23 PM
irella	bce46da	Today at 12:36 PM



The screenshot shows a Git GUI interface. On the left, a sidebar lists repository elements: demo, Changes, All Commits, Branches (master, 3↑), Remotes (origin), and origin (master). The main area shows a commit history with 'origin/master' selected. A dropdown menu is open over the 'demo : master' header, showing 'Local branches' and 'origin branches' with 'origin/master' highlighted. Below the commit history, a commit details view shows the author 'Mozzadrella', date 'December 16, 2017', and commit hash 'cc30f26f9bcb27fc45338961a3f09b269e'. A blue callout bubble points to the 'origin/master' option in the dropdown menu.

When we copied the repo by pushing it to remote, we also copied the pointer "master."





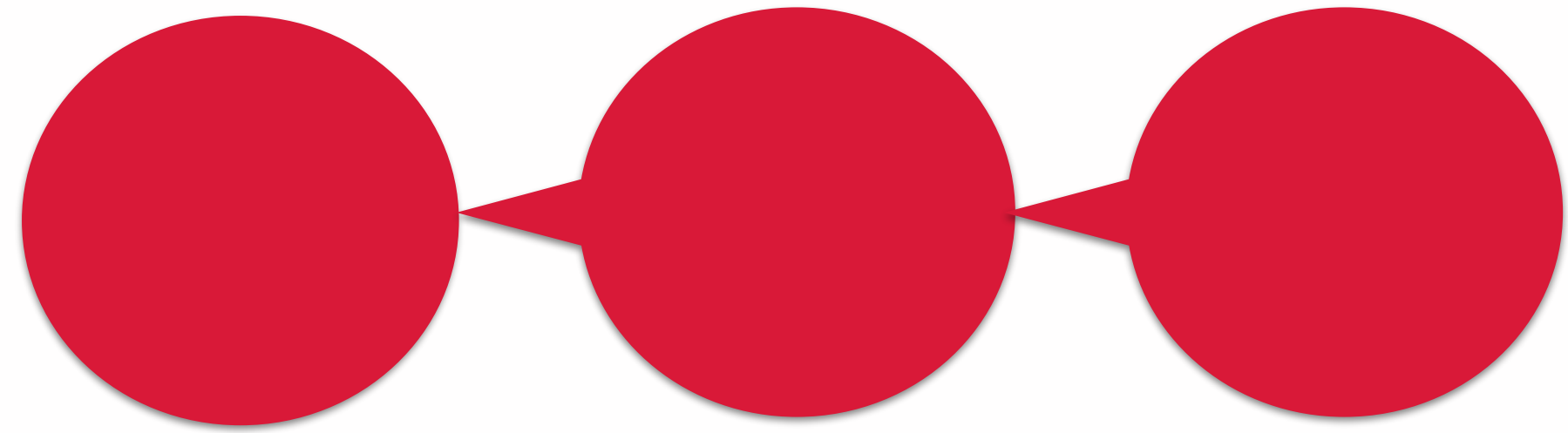
Since branches are bookmarks to commits...

in effect, every commit could be a branch.

When you commit, the active branch (bookmark) is updated.



Using branches in your terminal



Remember, branches are pointers to commits.

```
git show master
```

If we say 'git show master' we'll see the commit master points to.

```
commit cc30f26f9bcb27fc45338961a3f09b269ecd0931
```

```
(HEAD -> master)
```

```
Author: Mozzadrella <mozzadrella@github.com>
```

```
Date: Sat Dec 16 13:28:05 2017 -0500
```



Finding the active branch

'Git branch' will show you the branches in your project...

```
C02T40YZFVH4:demo  
mozzadrella$ git branch  
* master
```

and the "*" indicates your currently active branch.

If you made commits at that moment, the active branch would be updated to point to the new commit.



Creating a new branch

To summon a new branch, use 'git branch' and the new branch name. We'll call ours 'newbranch'

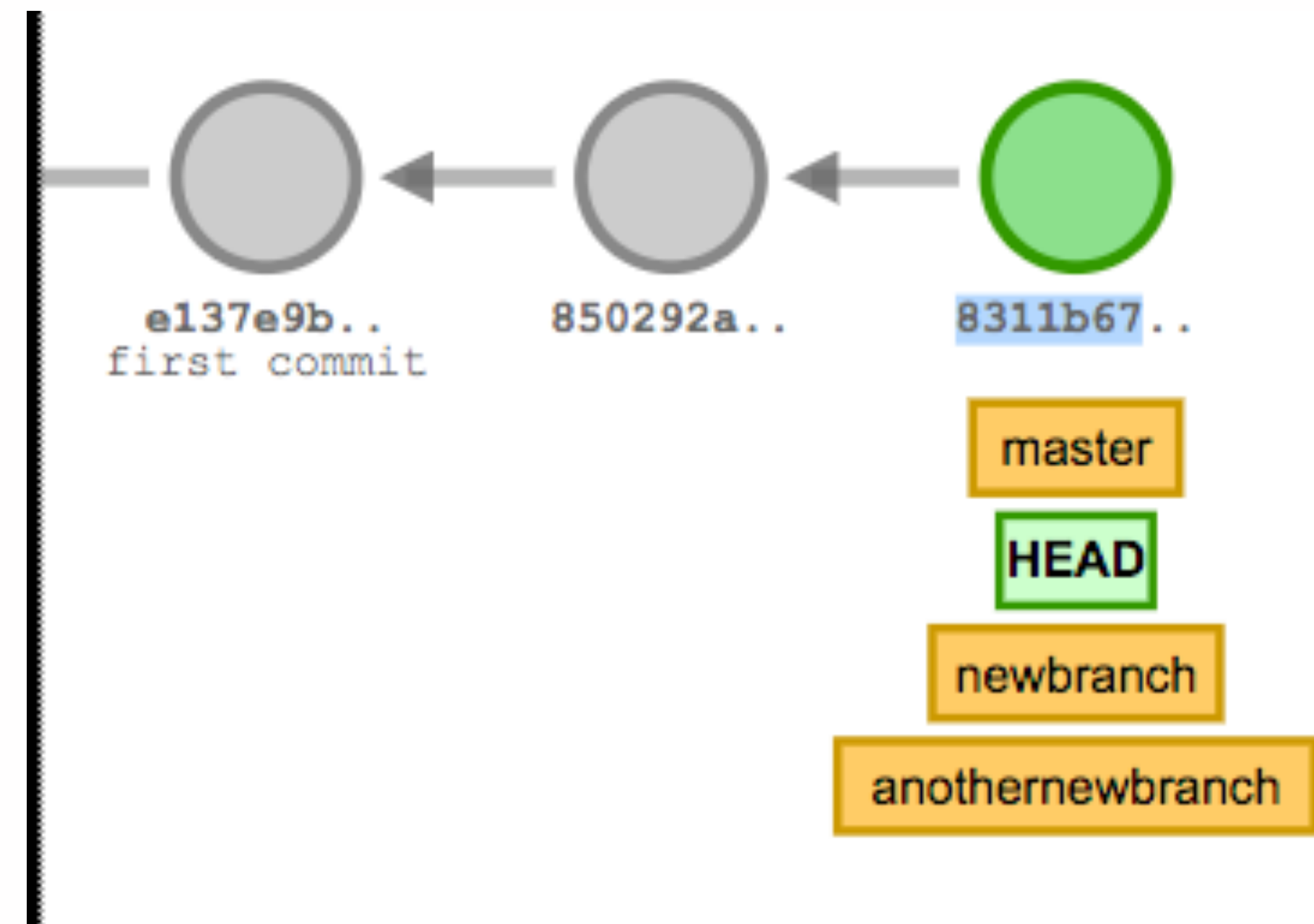
```
C02T40YZFVH4:demo mozzadrella$ git branch newbranch
C02T40YZFVH4:demo mozzadrella$ git branch
* master
  newbranch
```



Branches point back to the currently active commit

If we created 2 new branches from

8311b67



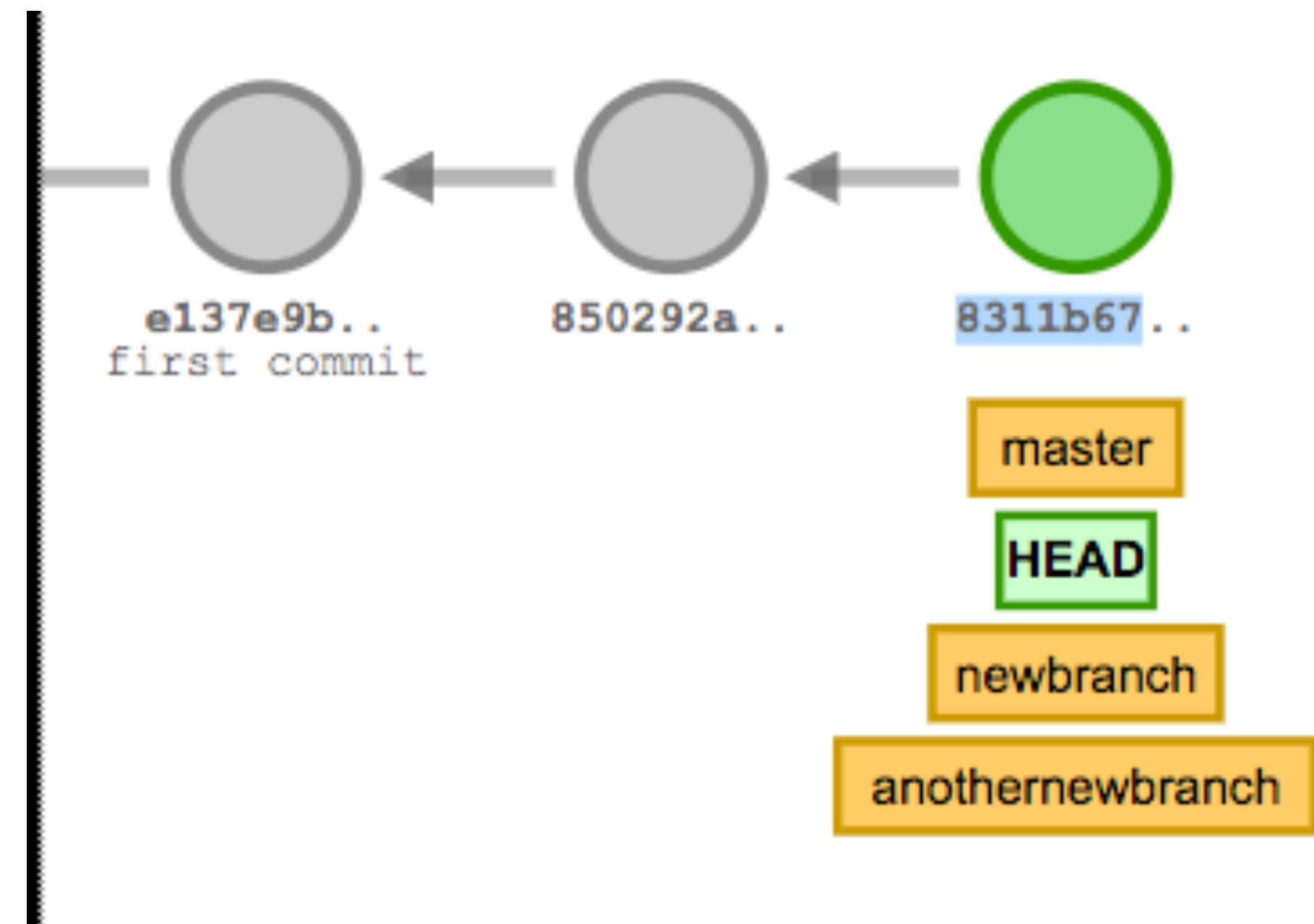
Branches point back to the currently active commit

If we created 2 new branches from

8311b67...

they would both point to

8311b67

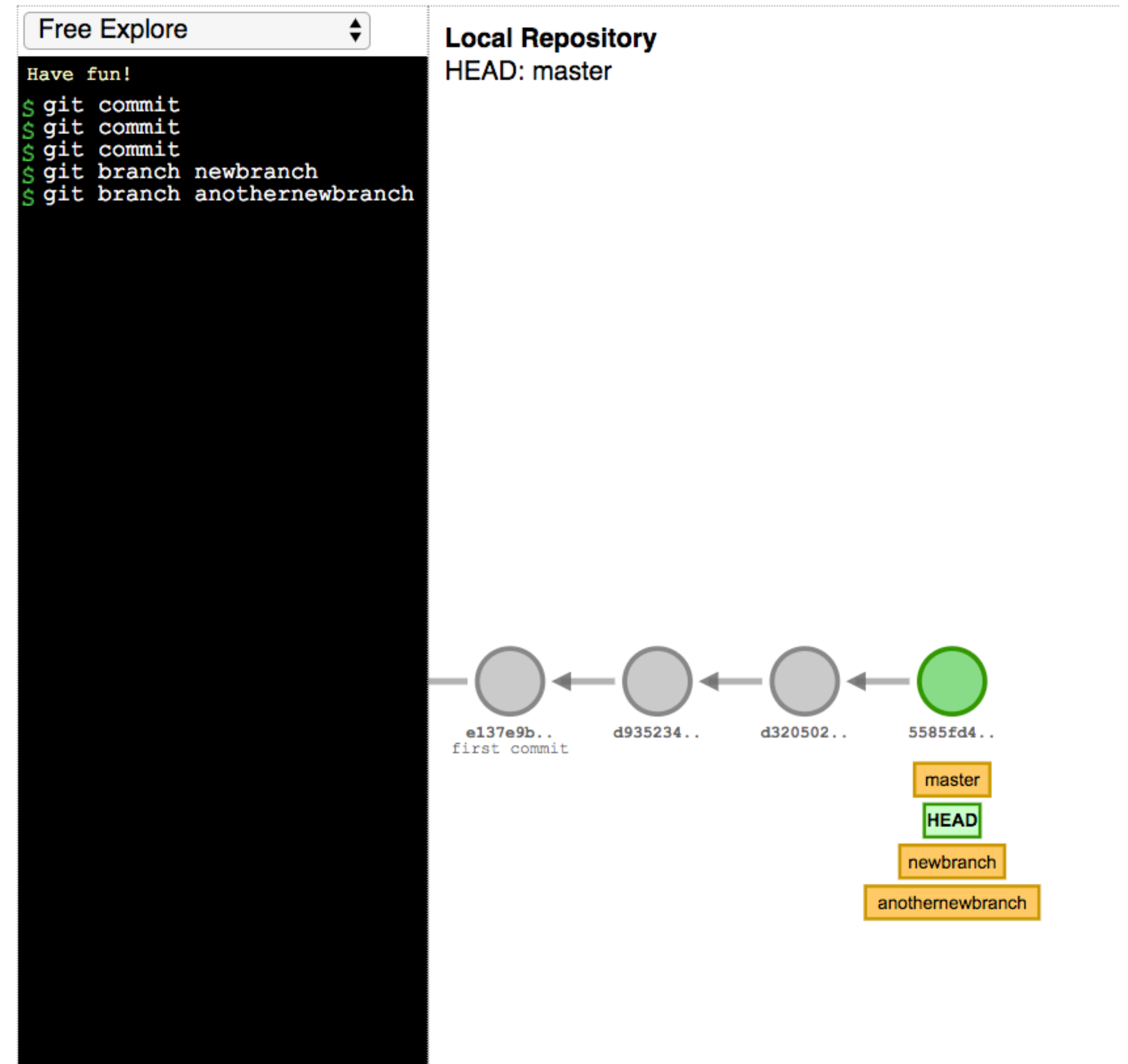


Branching from commits using references

`git branch <name>` creates a branch at HEAD

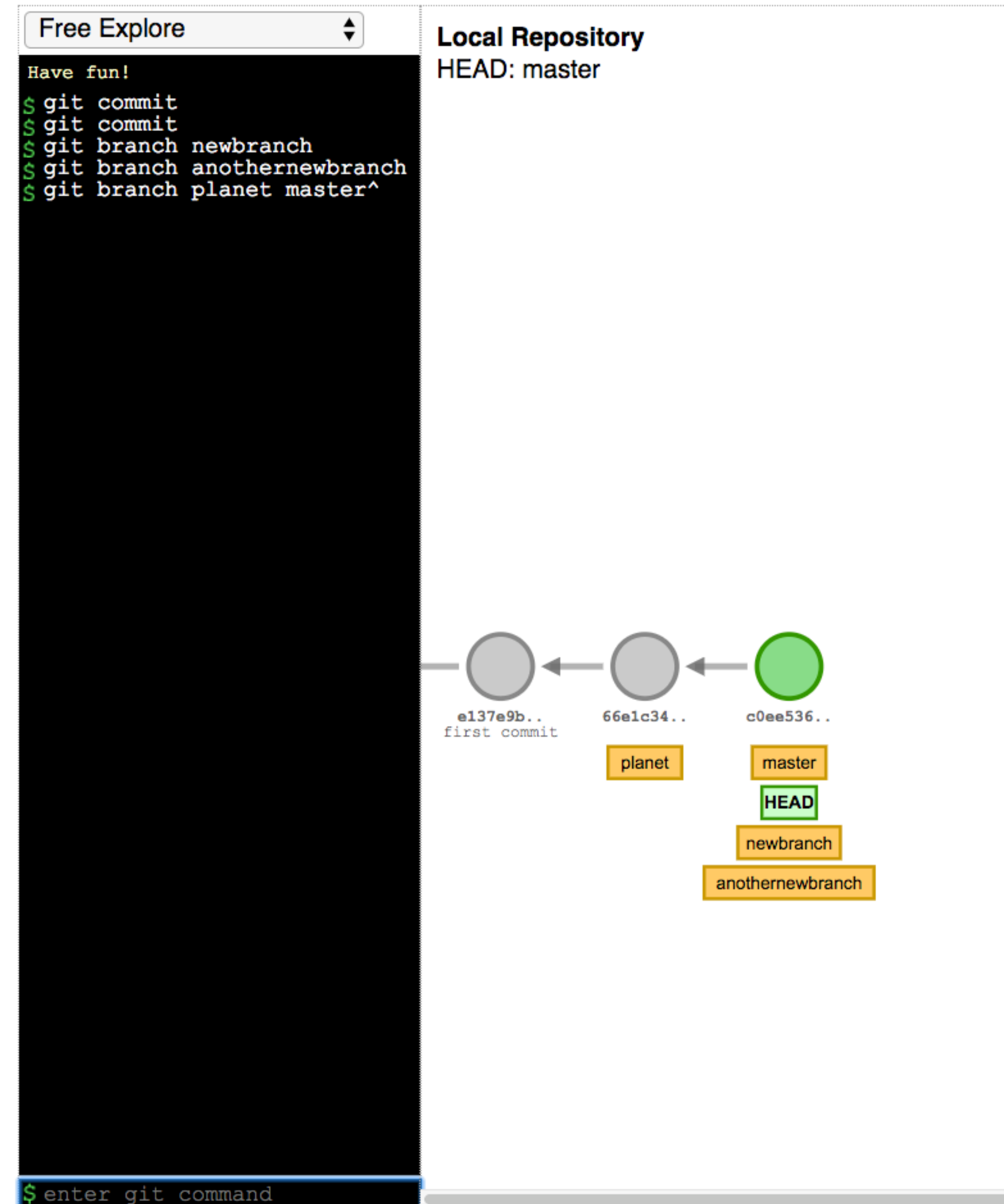
`git branch <name> <ref>` creates a branch at <ref>

<ref> can be HEAD, a branch name, a commit, or a commit-ish (e.g. HEAD[^] or master^{~3})



Or branch from previous commits

```
git branch planet master^
```




Recreate your assignment workflow



Free Explore

Have fun!

Local Repository
HEAD: master



e137e9b..
first commit

master

HEAD

\$ enter git command

The image shows a Git visualization interface. On the left is a black terminal window with the text "Have fun!" and a prompt "\$ enter git command". To the right is a white area representing the local repository. At the top right, it says "Local Repository" and "HEAD: master". A green circle represents a commit, connected to a vertical line. Below the circle, the text reads "e137e9b.." and "first commit". Below this, there are two boxes: an orange box labeled "master" and a green box labeled "HEAD", both pointing to the same commit.

Visit <http://git-school.github.io/visualizing-git/>

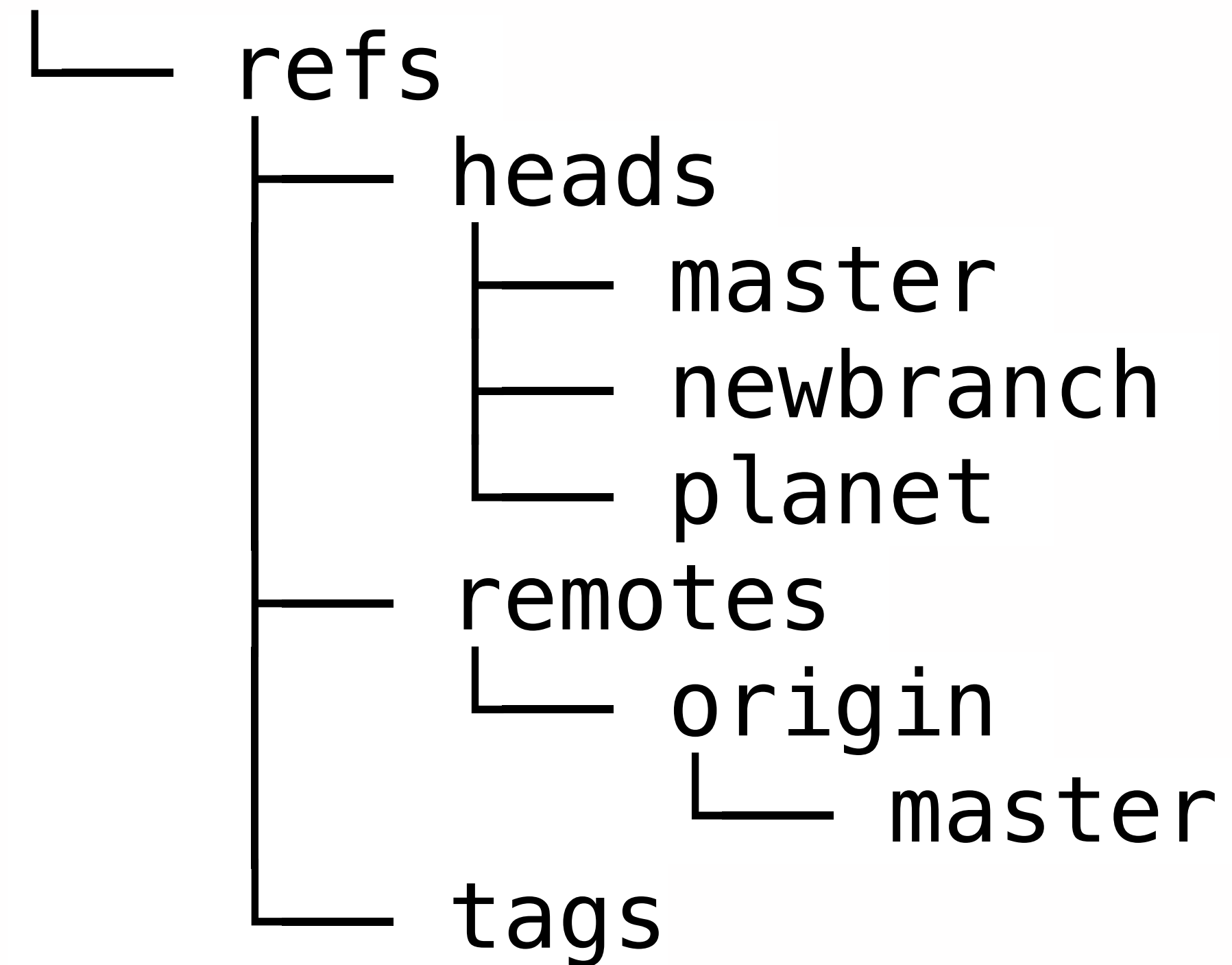


- You'll see the ability to make commits and branches.
- Choose an individual assignment from your course.
- Re-create the workflow you expect from your students to complete their assignment.
- Take a screenshot of your result from the visualization tool.



A final word on branches...

```
tree .git/refs
```



Those files contain the commit ID...

```
C02T40YZFVH4:demo mozzadrella$ cat .git/refs/heads/planet  
6c8bfc88bb440844f18a5e0a6ca885998b461bb7
```



So the implementation for branches is a file with a hash in it.



**Back to the world of
network activity.**

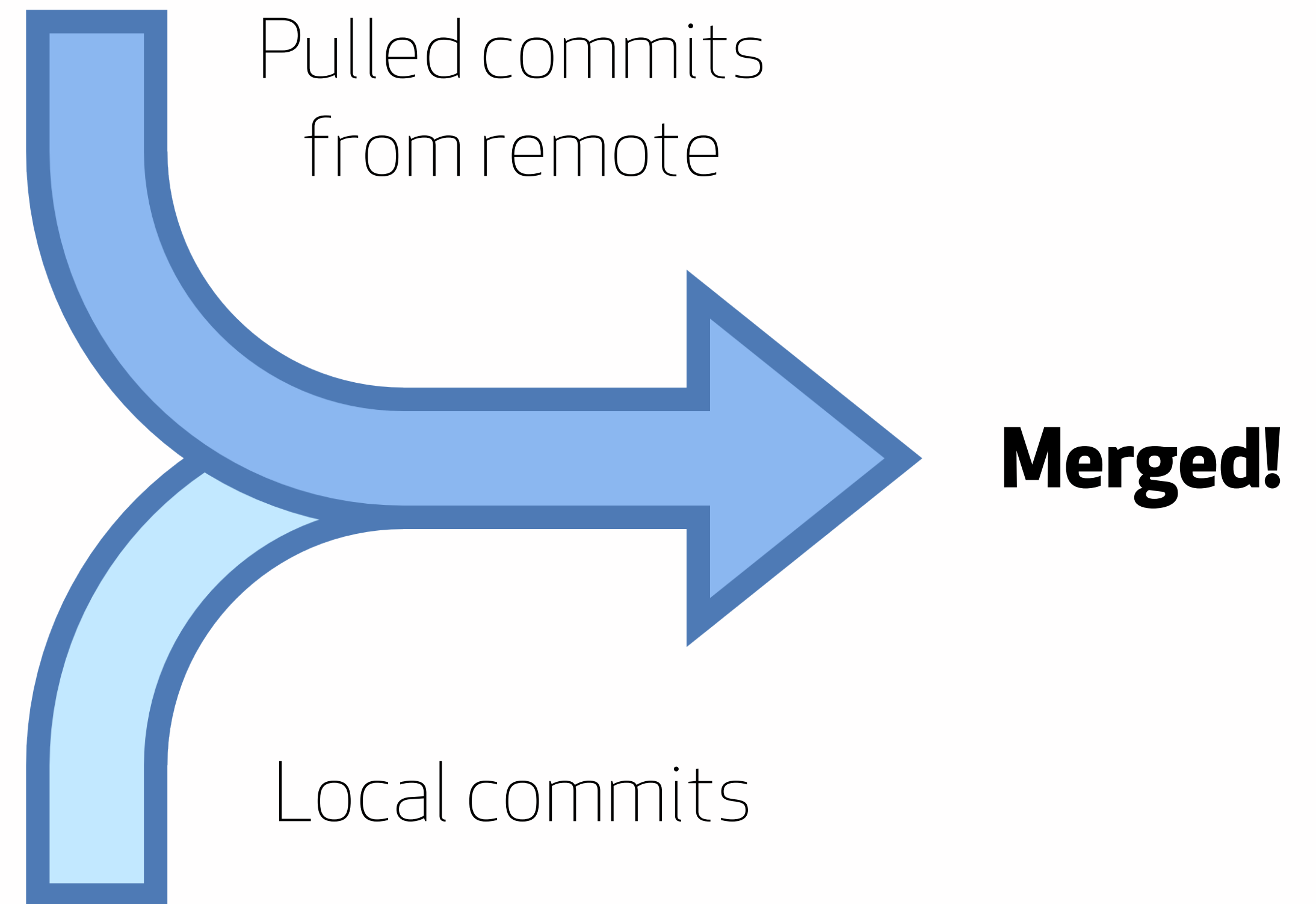
Pull = fetch + merge

Pull first fetches the commits and stores them locally.

Merge takes the two divergent commits, puts them together in the staging area and makes a new commit with two parents.

Merge updates the active branch to point to the new merge commit

You'll see the new commits reflected in your local project when you run "git log."



Watch what happens when we run "pull."

I'd like the latest commits on the branch that my active branch tracks.



```
git pull origin master
```



Watch what happens when we run "pull."

Thanks tracking branch!

`git pull origin master`



```
[C02T40YZFVH4:individual-work mozzadrella$ git pull origin maste
From https://github.com/mozzadrella/Module-1-Individual-Work
 * branch          master      -> FETCH_HEAD
Updating 1fa65cf..0869a21
Fast-forward
 readme.md | 4 ++++
 1 file changed, 4 insertions(+)
C02T40YZFVH4:individual-work mozzadrella$ █
```



To sum up, here are the commands with network activity:

```
git push
```

```
git fetch
```

```
git pull (fetch part, not merge)
```



Adding it all up



Wikipedia switcheroo

1. Create a repository on your desktop called 'planets' with a readme.md and commit it.
2. Choose a Wikipedia page about a planet of your choice.
3. Copy the first paragraph about the planet, paste it into your readme.md and commit it.
4. Create a new branch (`git branch <branchname>`) and switch to it (`git checkout <branchname>`).
5. In atom, add Wikipedia entry about **another** planet into that branch's readme.
6. Save, then commit, then close out of atom.
7. Does the file show your first planet? Or the second?



Managing student repositories



GitHub Classroom

The power of the GitHub workflow, scaled for the needs of students.

Automatically create student repositories, track assignments, and free up your time to focus on teaching.



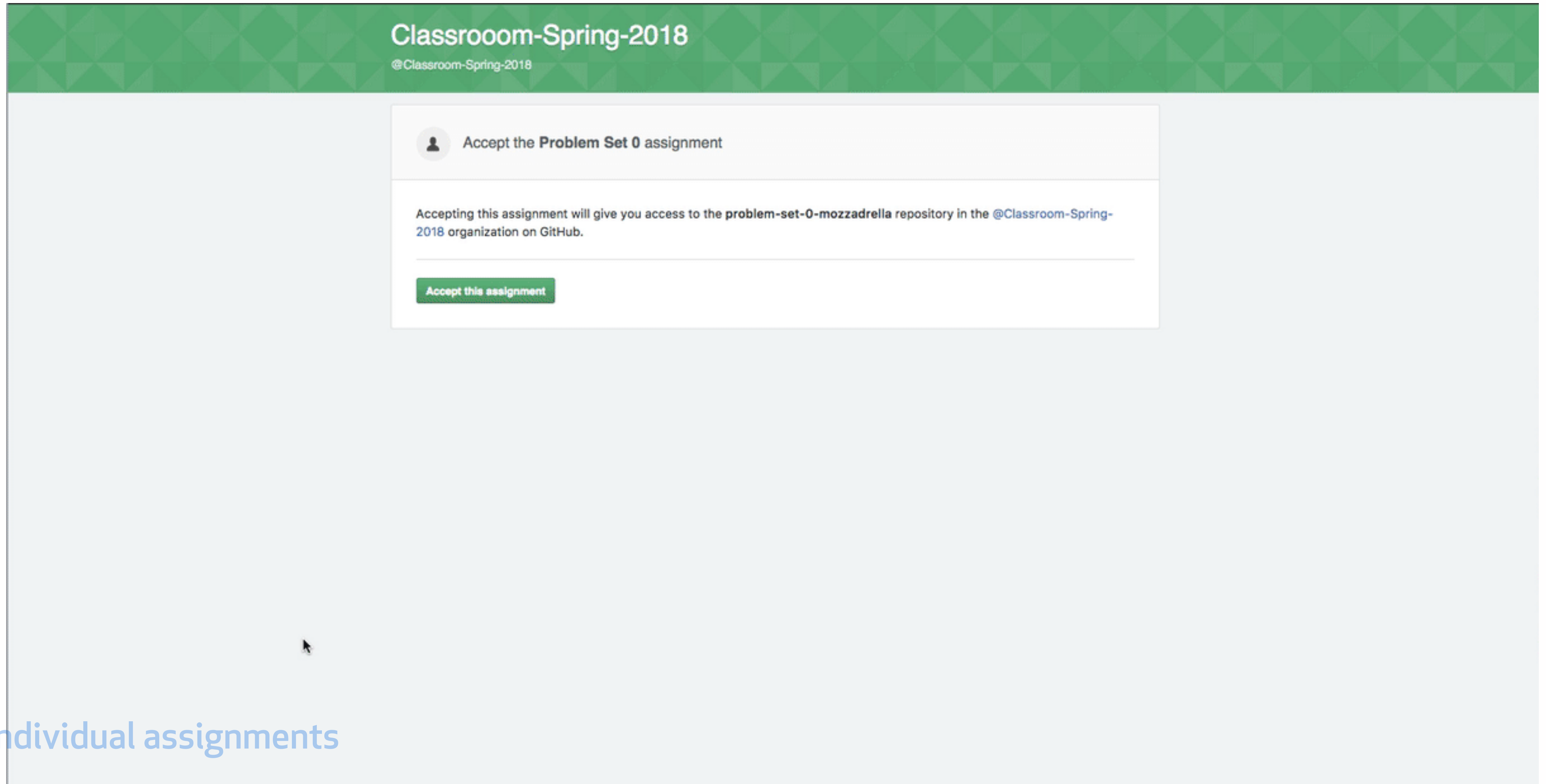
GitHub Classroom

Your course assignments on GitHub • classroom.github.com



Individual assignments

How a student experiences GitHub Classroom



The screenshot shows a notification in a GitHub Classroom interface. At the top, a green header bar contains the text "Classroom-Spring-2018" and "@Classroom-Spring-2018". Below this, a white notification box contains a user icon, the text "Accept the Problem Set 0 assignment", and a paragraph explaining that accepting the assignment grants access to the "problem-set-0-mozzadrella" repository in the "@Classroom-Spring-2018" organization. A green button labeled "Accept this assignment" is positioned at the bottom of the notification box.



How Dr. Root uses Classroom



I have 50 students
with one repository
per assignment.



Distributing assignments



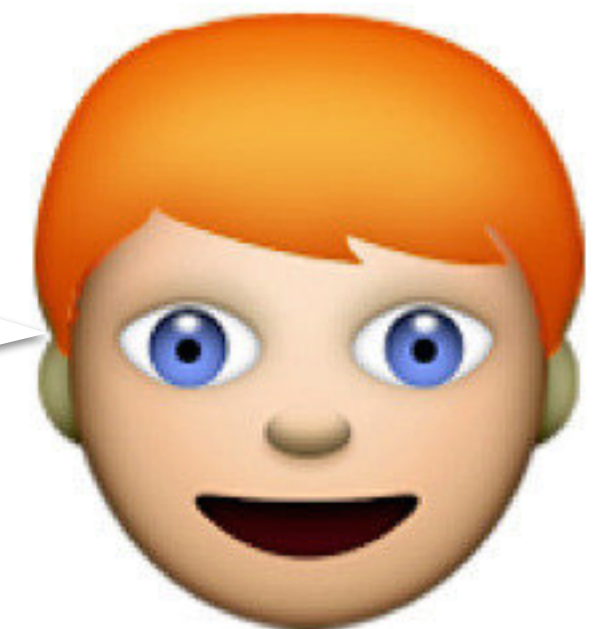
Here is the link to
Assignment 1,
class.

You'll all have
private
repositories with
all your starter
code.

assignment-1-
sarah



assignment-1-brian



assignment-1-alex



Deliver feedback in-context

- BowlingScore is actually a mixture of classes
• are Strike/Spun mark???

- is 'layer' intended
real misspell

- tons of variables, could have been aggregated better

- huge switch statements

- sledgehammer drawing!
numbers(letters)

- some APIs are awkward
(getX/getY/getZ separation)

Before



dondi 27 days ago

Your biggest issue is this one. As the explanation in Jenkins states, "Abbreviation in name 'twoRA' must contain no more than '1' consecutive capital letters." Bottom line—no consecutive capital letters. That implies an acronym or abbreviation, and proper readability prefers full spelling.

I will agree with this—I don't see how `twoRA` can be a name for the tally table, which is what this is. I mean, `tallyTable` would be a more descriptive name. Or perhaps `solutionTable`. Bottom line, cryptic names are not desirable.



Reply...

```
99         for (int i = 0; i < denominations.length; i++) {
100 +         twoRA[i][0] = new Tally(denominations.length);
101     }
102
103         for (int i = 0; i < denominations.length; i++) {
104             for (int j = 1; j < (amount + 1); j++) {
105                 Tally myTally = new Tally(denominations.length);
106 +                 myTally.setElement(i,1);
```



dondi 27 days ago

Two things wrong here: `myTally` starts 1 space too far (as the warning detail says), and there should be a space after the comma. The general rule with code punctuation is the same as written punctuation: no space before, but a space after. Keep that in mind and the code becomes more natural.



Reply...



Discussion:

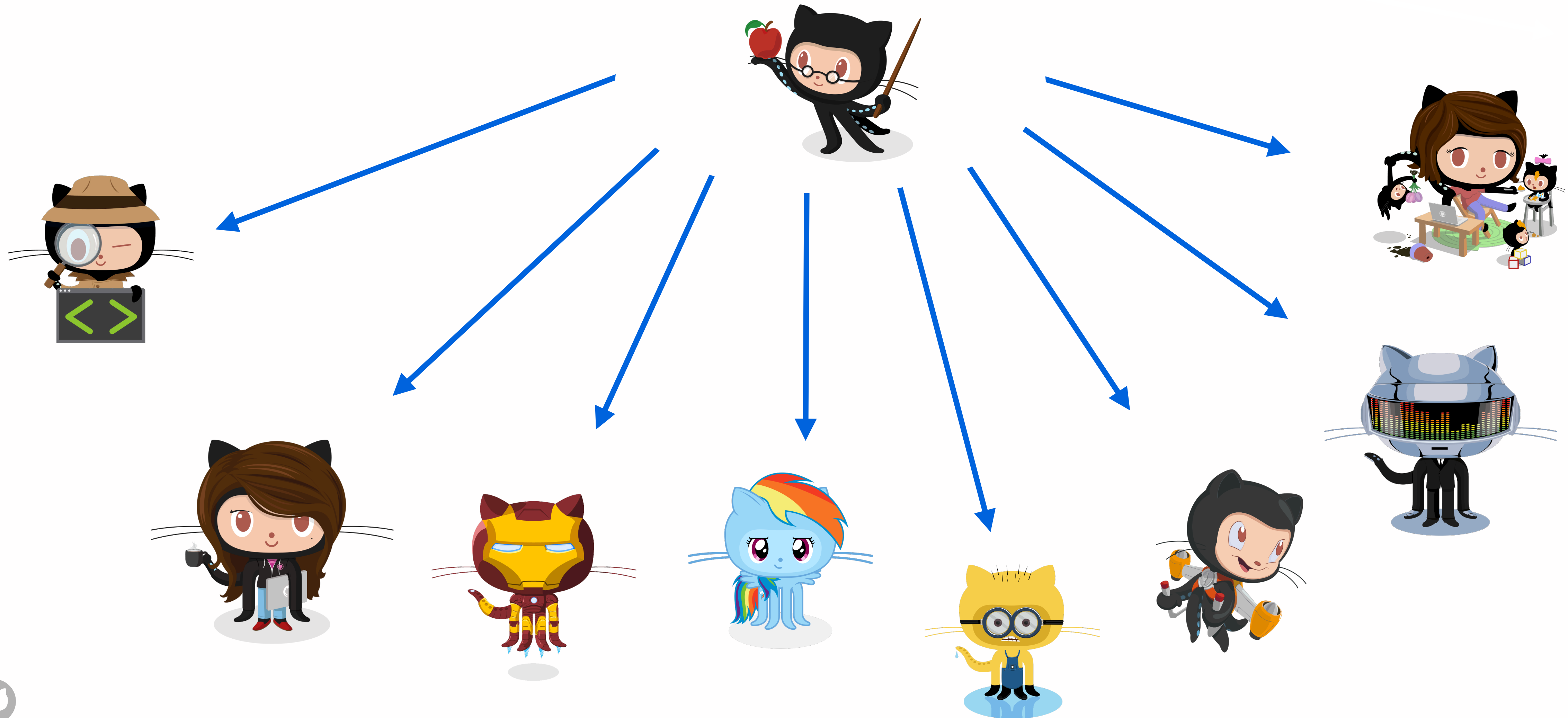
How might you imagine using this tool for your courses?



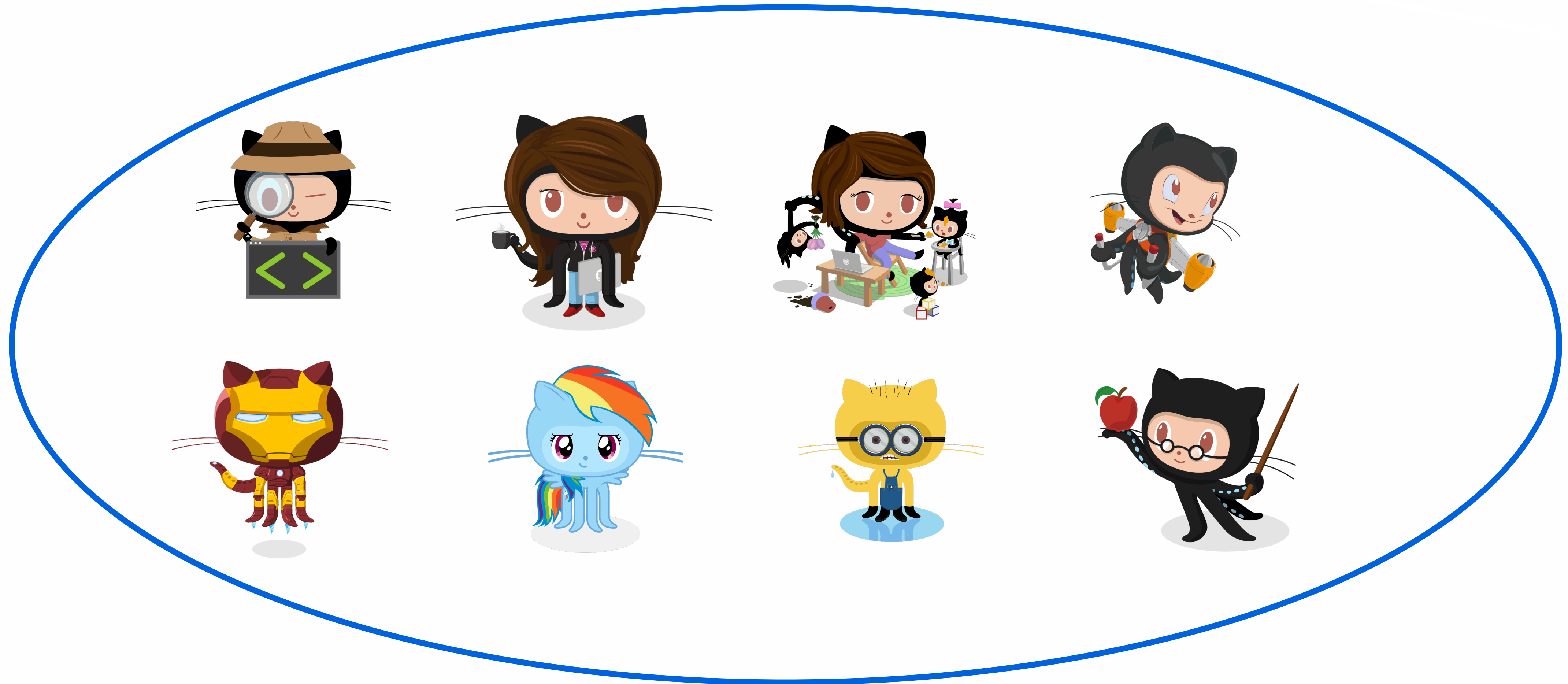
Setting up your Classroom



Individual accounts on GitHub



Organizations



Permissions levels

Owner

Administrator

Collaborator

Public access



Quick setup math

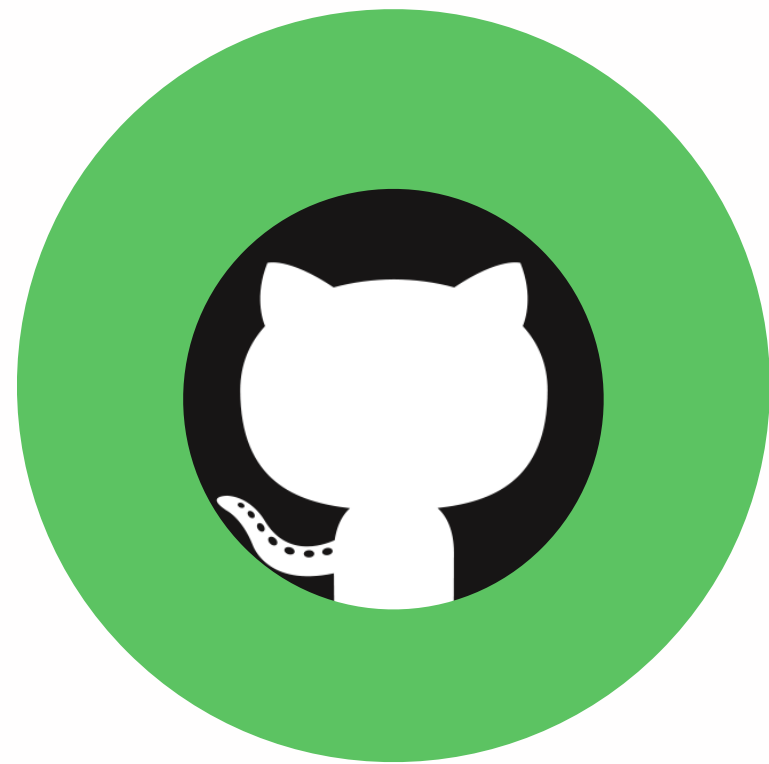
1 repository per student per assignment

1 organization per course per semester

All are free through GitHub Education



Set up a Classroom



- On [GitHub.com](https://github.com), create an organization for your course.
- Create repositories with assignment materials.




- Sign in to classroom.github.com
- Select your organization and authorize the application.



- Upload a roster of student identifiers.





Set up an Assignment

 **Individual assignment**
Each participant works individually on their own repository.
[Create an Individual assignment](#)

Create an individual assignment.

Set it to public or private.

Add your starter code from GitHub (optional)

-  Classroom-Spring-2018/Problem-1-Spring-CS1 Introduction to Parallel Programming class code
-  Classroom-Spring-2018/Problem-0-Spring-CS1 First programming assignment

After the deadline, GitHub Classroom will save the latest commit from each repo as a submission. Submission commits are viewable on the assignment page.

Add your Assignment's repository for starter code.

Deadline (optional)

Commit from each repo as a submission. Submission

Sun	Mon	Tue	Wed	Thu	Fri	Sat	17:00
31	1	2	3	4	5	6	18:00
7	8	9	10	11	12	13	19:00
14	15	16	17	18	19	20	20:00
21	22	23	24	25	26	27	21:00
28	29	30	31	1	2	3	22:00

Set a due date.

Give this to your students

<https://classroom.github.com/a/CBFrmelY>

Distribute assignment link.



Working with repositories



Discussion:

- Will you keep all course materials in a repository? Or just assignments?
- When will you expect students to commit?
- What sort of commit messages should they use?
- When do you want your students to push their code to GitHub?



Distributing assignments



Exercise:

- Create an organization with your course materials for one assignment in a repository.
- Create a Classroom, and an individual assignment.
- Send it to a colleague to accept.
- Take a snapshot of your teacher dashboard and post it to the forums.

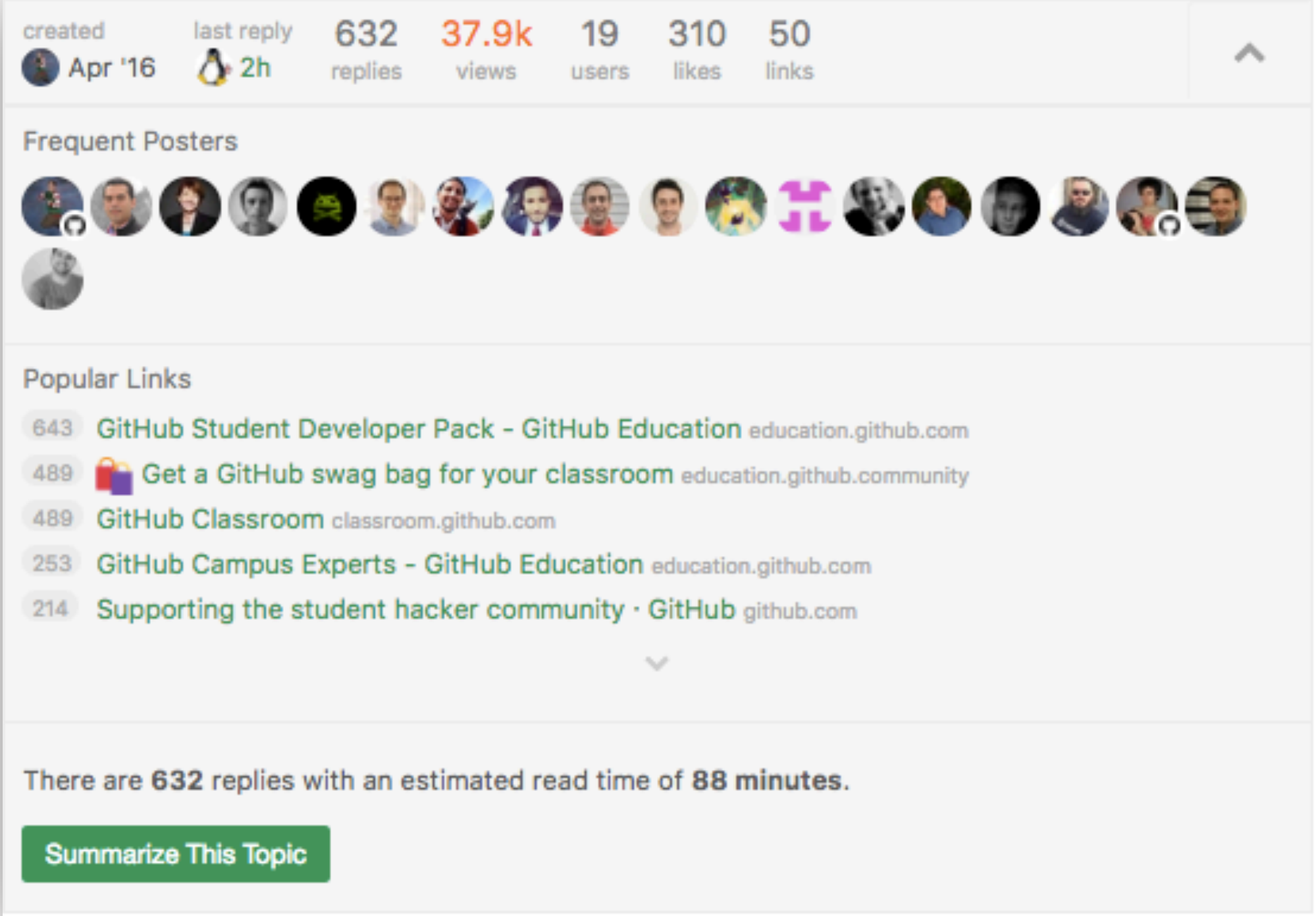


GitHub Education Community

Where veteran teachers share tips, tricks and scripts.

Best practices on everything from pull requests to assignment design.

Ask for help, get feedback and share your expertise: [education.github.com](https://education.github.com/community)



The screenshot shows a GitHub community discussion page. At the top, it displays statistics: 'created Apr '16', 'last reply 2h', '632 replies', '37.9k views', '19 users', '310 likes', and '50 links'. Below this is a 'Frequent Posters' section with a row of 18 user avatars. The 'Popular Links' section lists five links with their respective counts and destinations: '643 GitHub Student Developer Pack - GitHub Education education.github.com', '489 Get a GitHub swag bag for your classroom education.github.com/community', '489 GitHub Classroom classroom.github.com', '253 GitHub Campus Experts - GitHub Education education.github.com', and '214 Supporting the student hacker community · GitHub github.com'. At the bottom, it states 'There are 632 replies with an estimated read time of 88 minutes.' and includes a green button labeled 'Summarize This Topic'.

